CLAIMS

What is claimed is:

1. A method to efficiently design and implement a matched instruction set processor system, including:

analyzing and mapping design specifications of the matched instruction set processor into application components, each application component representing a reusable function commonly used in digital communication systems;

decomposing the matched instruction set processor system into interconnected design vectors, the interconnected design vectors are represented in Java programming language; and

analyzing and mapping the interconnected design vectors into specific hardware and software elements.

2. The method of claim 1, wherein analyzing and mapping design specifications further includes:

performing a behavioral analysis of the matched instruction set processor system to ensure compliance with the design specifications.

3. The method of claim 1, wherein analyzing and mapping design specifications further includes:

performing a requirement analysis of the design specifications of the matched instruction set processor system to generate a behavioral model; and

representing the behavioral model using application components.

4. The method of claim 1, wherein decomposing the matched instruction set processor system into interconnected design vectors further includes:

mapping the application components into corresponding architectural components.

5. The method of claim 4, wherein decomposing the matched instruction set processor system into interconnected design vectors further includes:

decomposing each application component into processing pipelines to satisfy system processing and timing requirements.

6. The method of claim 5, wherein decomposing the matched instruction set processor system into interconnected design vectors further includes:

decomposing each processing pipeline into design vectors, including functional design vectors and interconnect design vectors.

7. The method of claim 6, wherein decomposing the matched instruction set processor system into interconnected design vectors further includes:

using the functional design vectors to represent design information for at lease one functional aspect of the processing pipeline; and

using the interconnect design vectors to contain connectivity characteristics of the processing pipeline.

8. The method of claim 7, wherein decomposing the matched instruction set processor system into interconnected design vectors further includes:

providing in each design vector a binding header method, a run method, a Java virtual machine (JVM), a binding trailer method, and an invocation method.

9. A system to efficiently design and implement a matched instruction set processor system, including:

an application modeling unit to analyze and map design specifications of the matched instruction set processor into application components, each application component representing a reusable function commonly used in digital communication systems;

an architectural modeling unit operatively coupled to the application modeling unit, the architectural modeling unit decomposing the matched instruction set processor system into interconnected design vectors, the design vectors are represented in Java programming language; and

a realization mapping unit operatively coupled to the architectural modeling unit, the realization mapping unit analyzing and mapping the interconnected design vectors into specific hardware and software elements.

10. The system of claim 9, wherein the application modeling unit performs a behavioral analysis of the matched instruction set processor system to ensure compliance with the design specifications.

11. The system of claim 9, wherein the application modeling unit performs a requirement analysis of the design specifications of the matched instruction set processor system to generate a behavioral model.

12. The system of claim 9, wherein the application modeling unit represents the behavioral model using application components.

13. The system of claim 9, wherein the architectural modeling unit maps the application components into corresponding architectural components.

14. The system of claim 13, wherein the architectural modeling unit decomposes each application component into processing pipelines to satisfy system processing and timing requirements.

15. The system of claim 14, wherein the architectural modeling unit decomposes each processing pipeline into design vectors, including functional design vectors and interconnect design vectors.

16. The system of claim 15, wherein the architectural modeling unit uses the functional design vectors to represent design information for at lease one functional aspect of the processing pipeline.

using the interconnect design vectors to contain connectivity characteristics of the processing pipeline.

17. The system of claim 15, wherein the architectural modeling unit uses the interconnect design vectors to contain connectivity characteristics of the processing pipeline.

18. The system of claim 17, wherein the architectural modeling unit provides in each design vector a binding header method, a run method, a Java virtual machine (JVM), a binding trailer method, and an invocation method.

19. A machine-readable medium comprising instructions which, when executed by a machine, cause the machine to perform operations comprising:

analyzing and mapping design specifications of the matched instruction set processor into application components, each application component representing a reusable function commonly used in digital communication systems;

decomposing the matched instruction set processor system into interconnected design vectors, the design vectors are represented in Java programming language; and

analyzing and mapping the interconnected design vectors into specific hardware and software elements.